Technical Report No. 3

12
NW

PLATO Courseware Development Procedures[1]

William A. Mahler[2]
A. Lynn Misselt[3]
Richard M. Schell[4]
and
Donald L. Alderman
Educational Testing Service
Princeton, New Jersey 08540

September, 1976

APPROVED FOR PUBLIC RELEASE

DISTRIBUTION UNLIMITED

Sponsored by

Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, Virginia 22209

ARPA Order No.: 2651
Contract No.: MDA 903-74-C-0290

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 10	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 PLATO Courseware Development Procedures.		5. TYPE OF REPORT & PERIOD COVERED 9 Technical Report, No. 3
		6. PERFORMING ORG. REPORT NUMBER MDA 903-74-C-0290-10
7. AUTHOR(s) 10 William A. Mahler, A. Lynn Misselt, Richard M. Schell, Donald L. Alderman		8. CONTRACT OR GRANT NUMBER(s) 15 MDA 903-74-C-0290, ARPA Order-2651
9. PERFORMING ORGANIZATION NAME AND ADDRESS Educational Testing Service Princeton, New Jersey 08540		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4W10-0002
11. CONTROLLING OFFICE NAME AND ADDRESS DARPA 1400 Wilson Blvd. Arlington, Virginia 22209		12. REPORT DATE 11 September 1976
		13. NUMBER OF PAGES 91
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) SAME 12 91p.		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

14 ETS-TR-3

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Authoring
Computer Assisted Instruction
Curriculum Development
Instructional Materials
Production Models

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is an exploratory study of methods for the preparation of computer curricular materials. It deals with courseware development procedures for the PLATO IV computer-based education system and draws on interviews with over 100 persons engaged in courseware production. The report presents a five stage model of development: planning; production; review; validation and documentation; and implementation. This model then serves as a basis for the discussion of various group structures which range from the → next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

122 250

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

cont.

independent developer to team members with special skills. Features of the PLATO system did affect the group productivity as well as quality of lesson material: these influences, both beneficial and detrimental, appear to be important factors to consider in the organization and management of courseware development groups. A major objective of this study, beyond the description of methods for development, was to set forth those characteristics that might distinguish among groups and help in personnel selection. Therefore the report includes a discussion of group and individual characteristics, and suggests further lines of research on selecting and training developers. A summary of this study and extensive recommendations for further work conclude the paper.

CONTENTS

CONTENTS (Cont.)

The development of instructional materials, or "courseware," constitutes a major component in the use of computers or other media for instructional purposes (Carnegie, 1972; Anastasio and Morgan, 1972; Hunter, et al., 1975). This paper reports on a study of procedures used by a number of groups to develop courseware for the PLATO[5] system. The common elements in the steps taken to develop usable courseware, the personnel considerations involved in this task, and factors which exerted a strong influence on the process of courseware development are described. The discussion should help to clarify the complex task of courseware development and stimulate further efforts to refine the process.

There have been many publications which propose courseware development procedures (see reviews by Gagné and Rohwer, 1969; Merrill, 1971; Schutz and Baker, 1971; Glaser and Resnick, 1972; McKeachie, 1974). A few articles have included descriptions of the operations of courseware development efforts (Yelon, 1973; Reed, Ertel, and Collart, 1974; Casheil, Lent, and Richardson, 1975; Diamond, et al., 1975), while others have described the research or theory on which procedures were based (Bruner, 1966; Anderson, et al., 1969; Gagné, 1970; Atkinson, 1972; Atkinson and Paulson, 1972; Merrill, 1972; Merrill and Boutwell, 1973). Further articles have dealt with general limitations of the courseware development process (Locatis, 1973), the political

contexts of these efforts (House, 1974; Fraley and Vargas, 1975), and methods for evaluating the different procedures (Felker and Shettel, 1975; Smith and Murray, 1975). In addition, there have been publications dealing with the unique aspects of instructional development for computer-based systems (Bunderson, 1973; Hillelsohn, 1974; Reed, Ertel, and Collart, 1974; Broderick, 1975; Eshenbrenner and Lamos, 1975). However, there have been no previous attempts to study a variety of procedures. Without close examples to follow, this study could take only a broad, preliminary look at the spectrum of courseware development procedures used with the PLATO *system* and the factors which influence them. Its purposes include the following:

1. Synthesize the procedures used by people working with the PLATO system through the 1974-75 academic years into descriptive models of the stages that instructional lessons went through as they were developed and of the personnel structures that were used to accomplish this work.
2. Raise questions about the determinants of the <u>selection</u> and <u>productivity</u> of these procedures.
3. Recommend hypotheses and methodologies for further research on courseware development procedures.

No attempt has been made to relate this study to judgments about the effectiveness of the courseware or the individuals involved. Therefore, final judgments about the procedures would be premature. Hopefully, other researchers will use the recommendations of this paper in collecting the data necessary to choose among the various courseware development procedures described.

The data for this study were collected between June, 1974 and March, 1975 through interviews with 122 people who were involved in the development of courseware for the PLATO system. With few guidelines concerning the specific information to be sought, questionnaires, on-line data records, and daily logs were found to yield incomplete descriptions of procedures, which were rapidly changing. Interviews (see appendix) provided the flexibility to ask questions appropriate for each interviewee and yet to cover a broad range of topics. The people to be interviewed were selected on the basis of their affiliations with certain groups of developers. All projects which were funded by the agency which funded this study were included. In addition, seven groups supported as part of a national field trial for PLATO in public education, three university-based courseware development groups and three support groups, organized to support development efforts of other groups, were included. See Table 1 for a listing of the groups, the dates of their interviews, and the number of people interviewed. More information about the data collection procedures can be found in the final section of this paper.

Insert Table 1 about here

In order to put the discussions of procedures and their determinants into context, the next section describes the PLATO system. The following section deals with the stages of courseware development and the characteristics of the courseware which affect its development. The organization and management of courseware development efforts and their influences on the process are then described. The report concludes with a combined summary and recommendations for further research.

THE PLATO SYSTEM

Development of the PLATO system has been carried on continuously at the University of Illinois since 1959. The first product of this research was PLATO I, a single terminal, prototype system using Illiac I, an early electronic computer. Subsequent developmental stages, PLATO II and PLATO III, were multi-user systems of increasing size and sophistication. The current system, PLATO IV, became operational in 1971 and came into extensive use in 1972 (see Lyman, 1975 for historical highlights and a list of publications describing the development of the PLATO system).

Features of the PLATO IV System

The PLATO IV system has several distinct hardware features which directly and indirectly influence courseware development procedures (Wood, 1975). These include:

1) Plasma display panel terminals which can selectively display and erase standard alphanumeric or user-designed characters and line drawings at any screen location. These terminals can also support rear projection microfiche image selectors, touch-initiated input, random access audio devices, and a variety of other peripherals. The terminals can be used either by developers to create new software and courseware or by students to interact with the courseware which has been developed (Stifle, 1974).
2) A large central computer to which 1,000 terminals can be connected. The extended core storage or ECS, an integral part of this computer, allows rapid response to input from users at terminals (Stifle, 1972).
3) A telecommunications network which uses microwave transmission, voice-grade phone lines, and other

devices to connect remotely located terminals to the central processor (Sherwood and Stifle, 1975).

In addition to the unique collection of hardware involved, another distinctive and important component of the system is the TUTOR authoring language (Tenczar, 1975; Sherwood, 1974). Development of this language began at the University of Illinois in 1967 (Avner and Tenczar, 1969) and has been refined continually on the basis of the experience and needs of the users (Tenczar, 1975). Its key features include:

1) Computational features, such as implied multiplication and superscripts for exponents, that closely mimic natural algebra, plus compilation of all calculations into machine code for rapid execution.
2) Numerous commands which enable developers to take advantage of the graphic capabilities of the plasma display panel terminals.
3) Extensive answer-judging capabilities.
4) A large selection of branching commands to assist in individualization of instruction.

There are five discernible categories or levels of software in use on the PLATO system. These are:

1) System software for running PLATO programs, including the display formatter, the TUTOR language compiler (or condensor), a memory manager, and input-output support.

2) System software for supporting and facilitating computer-based education. These include course rosters or "records" through which instructors give access to the system, "routers" for sequencing students through instructional segments or "lessons," and routines for collection of data which summarize students' progress through the lessons.

3) Systems software for assisting in the development of lessons. Among the most important features provided are: the text editor which includes powerful tools for the creation of displays; AIDS - an online guide to the TUTOR language and PLATO system; special inter-terminal communications support which allow a user to view another user's lesson material, as well as to "talk" to the other user by displaying typed messages simultaneously on both screens.

4) Programs provided by the user for inter-lesson routing, student data collection and analysis, and special online administrative routines. Such support programs are generally written to enhance or extend the capabilities of the software mentioned in 1 and 2.

5) "Lessons" or instructional programs which contain the specific information and displays which are presented to the student. Students using a condensed lesson do not see the actual TUTOR code which operates the displays which they see. Various types of lessons are described in a later section.

The last two categories of software are called "courseware" in this paper because both instructional "lessons" and the support programs in category four had to be developed by the courseware development groups. The effects of developing support programs are discussed in a later section.

Courseware has been developed in numerous subject areas and for levels of instruction ranging from preschool to graduate education (Lyman, 1975b). These efforts can be roughly divided

into three time periods, as follows: prior to 1967-68, 1967-68 to 1971-72 and 1971-72 to 1975-76. During the first period, the PLATO system itself was being developed through versions I, II, and III with primary funding from the U. S. Department of Defense. Courseware development was necessarily limited to a few organized efforts at the University of Illinois. A rudimentary language for courseware development, called CATO, was introduced in 1965, but developers had to be quite conversant with other computer programming languages. One research project on courseware during the period was funded by the U. S. Office of Education (Easley, 1968). Some efforts were the work of individuals without outside support or with small grants from various sources.

Two events in 1967-68 affected courseware development efforts on PLATO. First, the Computer-based Education Research Laboratory (CERL) was established to continue research and development on the system and to provide operational support for system users. Second, the TUTOR language was introduced as a means for developing courseware without an extensive background in computer use. These events led to the beginnings of the first four groups included in this study and many other efforts. Most of these "groups" were initially just individuals who became interested and started to work on their own. An additional impetus to expansion was the infusion of funding from a variety of sources, including the National Science Foundation; the U. S.

Department of Defense; the State of Illinois; the U. S. Department of Health, Education and Welfare; the National Institutes of Health; the Metropolitan Museum of Art; the Ford Foundation; the U. S. Agency for International Development; the Control Data Corporation; Owens-Illinois, Inc.; and others. Some of these funds were granted for specific courseware development efforts, but much of the money was devoted to the development of the system which was then available at no charge to anyone with the initiative to get involved.

In 1971-72, the PLATO IV system with its increased capabilities and availability, as already described, was introduced. At approximately the same time, multi-million dollar projects to test the instructional, technical, and cost effectiveness of PLATO over a period of four and a half years were funded by the National Science Foundation and by the U. S. Department of Defense through its Advanced Research Projects Agency. Both projects included significant commitments that courseware would be used and evaluated, although specific courseware development mechanisms were not prescribed. Later, additional funds had to be appropriated for courseware development because there proved to be insufficient support available for these efforts. Most of the groups included in this study received support from these two projects and hence may be a biased sample. An effort was made, therefore, to include three other groups even though the wide variety of small efforts could not be included.

At the same time that CERL received these large grants, the Educational Testing Service received separate contracts to conduct an independent evaluation of the NSF project and to facilitate instructional research related to the ARPA project (Anastasio, 1972; Alderman and Mahler, 1973). The later contract included a commitment to study the courseware development procedures used with the PLATO system. This paper is the report of that study; reports on other aspects of these projects will be issued separately.

COURSEWARE

Stages of courseware development

In order to provide a focus for the following discussions of the determinants of courseware development procedures, this section begins with a generalized description of the procedures under investigation. This description shares many characteristics with the numerous general instructional development systems which have been proposed or described (for example, Gagné, 1962; Glaser, 1966; Smith, 1966; Biggs, 1970; Schutz, 1970; Gerloch and Ely, 1971; Johnson and Johnson, 1971; Kemp, 1971; Popham, 1971; Baker, 1973; Eisele, 1973; Pents, 1973; Wallen, 1973; Gagné and Biggs, 1974; Gow and Yeager, 1975; Kozma *et al.*, 1975; McManus, 1975).

There is also correspondence with a proposed format for reporting on PLATO lesson development, which includes the stages of general planning, preliminary design, final design, formative evaluation, implementation, summative evaluation, and maintenance (for a description of these stages, see Avner, 1973). The description given here had to be made more flexible and less prescriptive than previous descriptions because of the wide variety of approaches used by PLATO developers. Some groups have adopted rather elaborate and structured procedures, while others have proceeded on an ad hoc basis. Figure 1 is intended as an abstract summary and not as a description of any particular group or approach

----

Insert Figure 1 about here

Planning. It is possible to start with any of the three activities which make up the planning stage (see Figure 1). In many cases, the content or behavioral objectives were defined, an appropriate instructional strategy was selected, and then a basic program structure was created or adapted from an existing lesson. In other cases, the developer began because of well-formulated pedagogical reasons with a commitment to a particular instructional strategy, such as drill and practice or simulations, then identified appropriate content objectives, and finally specified the program structure. In rare cases, a developer chanced upon

an interesting program, usually a game, and then developed objectives and selected content from a different subject area to fit that format.

The depth and breadth of material being planned varied across groups and time. Four of the thirteen production-oriented groups started with the idea of developing curricula for single courses or multi-year educational programs. Five groups and some members of another group started with individual lessons and then later planned entire curricula. The other members of the split group and another group planned individual lessons which were never expanded into entire curricula. Two other groups decided to develop lessons which fit into an existing curriculum, but once the topics had been determined, each lesson was planned independently.

The formality of the planning process ranged from the personal thoughts of an individual to written proposals which were reviewed by a committee. Likewise, refinements to plans were based on personal communications between developers in four of the thirteen production-oriented groups, while the other nine groups had more formal mechanisms, such as revised proposals or group meetings. It was found that in addition to revising the plans for individual lessons, twelve of the thirteen groups made changes in their overall goals. Such changes included new target audiences, reductions and expansions in the breadth of coverage, reductions and expansions in the depth of coverage, reductions and expansions in the number of institutions to be served, and changes from a research orientation to a production orientation.

Production. As seen in Figure 1, production is the center of the entire courseware development process. It entails the fabrication of the design specifications of the planning stage and the modifications suggested by the review and validation stages. It is complete only when the lesson is ready for operational use by students. Production was largely carried out by individuals who were sufficiently proficient with the TUTOR language to program the lessons under development, although non-programmers frequently furnished suggestions and ideas.

Since production involves all coding and debugging of lessons, it was generally the most time-consuming stage of the process. In general, the amount of production work was reported to be inversely related to the amount of detail in the design specifications and directly related to the number of revisions caused by criticisms, the discovery of omissions, and the creation of new ideas from other stages. Although there is no conclusive evidence, it appears that production time was also affected by the complexity of the lessons, the use of intra- and inter-lesson connections, the use of complicated graphic displays, and the need for special data collection routines.

Almost all production work by the vast majority of the developers was done while using a PLATO terminal. While a very few people went to the extreme of writing out their TUTOR code beforehand, most people hadonly a general notion of the code they

were about to produce when they sat down at a terminal. Some people developed flowcharts, particularly for complex support routines, before they began programming on-line. Most developers used printouts of their existing code when making revisions, partly because the limited size of the PLATO screen does not allow all parts of the program, which may have many interactions, to be viewed simultaneously. One reason for the great amount of on-line work was the ready access of the system beginning in 1972-73. Another reason is the availability of many on-line aids and the capability of testing out new sections of code immediately. One developer contended that since interaction with computer-based courseware is very desirable and beneficial, it is only natural that experienced developers would use a great deal of on-line interaction in their own work habits.

Review. Not only were a variety of people asked to critique lessons, but their opinions on the quality of the instructional strategy, the efficiency of the coding, and the accuracy of the subject matter content were all the targets of reviews (see Table 2). In some groups this process was quite formalized and was supervised by someone other than the primary developers (Francis, Goldstein, and Sweeney, 1975; Essex, 1975); in other groups each author informally solicited comments according to personal preference. While personal, face-to-face review by a peer was the most common form of review, some groups

found it beneficial to have critiques given in written form because of the great personal attachment that some authors had to their lessons (Francis, Goldstein, and Sweeney, 1975; Essex, 1975). In addition to critiques by peers and experts, the review stage included initial student testing, which was often done with fragments of lessons as well as complete lessons. While some groups collected and analyzed on-line data and written comments from these first students, most information from initial student testing was collected by watching the first few students and observing problems as they occurred. This procedure also allowed the developer to furnish assistance if the student could not proceed.

As Table 2 indicates, other developers, who were usually from the same group as the primary developer of a lesson, were the most common reviewers. Most groups also did some type of student testing before making a lesson generally available or using it as part of the regular instructional program. The few external experts who were used as reviewers usually looked at only the overall goals of the project with specific lessons as examples. The content and instructional design received about the same amount of review and were often reviewed simultaneously. The actual TUTOR code received less review, as would be expected, and such reviews usually focused on identifying potential "dead-ends" or other programming "bugs."

Insert Table 2 about here

Validation and documentation. Although some PLATO lessons have been validated and documented, many have not yet reached this stage of development or were inteded for private use. Proper validation and documentation can be very useful in encouraging the widespread usage needed to justify the substantial expenditures of time and resources used in the previous stages of courseware development. Avner (1973) encouraged that documentation and validation be built into the entire process of lesson development and not simply added on the end. He advocated an iterative approach which creates increasing specification of the target audience, objectives, instructional strategies, and validating evidence as the lesson itself is developed. Avner (personal communication) suggested five types of validating data which can be easily collected by the system: student time spent, degree of interaction, lesson difficulty, anticipation of student needs, and relation to external measure of achievement, attitude, or behavior.

Most of the documentation of PLATO courseware that exists thus far focuses mainly on descriptions of the content and instructional design (see Lyman, 1975 for a list of publications about PLATO). Statistics on usage, reactions from students and instructors, and records of student response data collected by

the system are also included (for example, Alpert and Jordan, 1976). Published comparisons of PLATO courseware and other forms of instruction have been written by McKeown and Lenehen (1974), McKeown (1974), Bennett (1975), Dare et al. (1975), and Montanelli (1975). In addition, there are reportedly other internal or unfinished studies of the effectiveness of PLATO courseware.

Implementation. The three tasks of the implementation stage are quite distinct (see Figure 1). Maintenance of lessons includes the monitoring and analysis of data being continuously collected, making minor programming changes in response to system changes, and adapting materials to fit the needs of new users, for example, changing the sequence of certain lessons or the type of mathematical notation used. The groups involved in the NSF project were carrying out such tasks, but most other groups were not yet providing such services to other users. Updating materials implies substantive changes which should be undertaken only by a qualified professional, presumably with the permission or cooperation of the original developers. Such changes could reflect new knowledge in the subject area, system capabilities which were either unavailable or simply unused, or improved instructional strategies. None of the groups were yet updating lessons at the time of this study. Groups were only disseminating their courseware upon request or part of the NSF-sponsored field test. Such activities included demonstrations, training sessions,

consulting on proper usage, and other efforts to encourage use by faculty at the participating schools. In addition, there were people who worked at the schools and were responsible for keeping the PLATO operation functioning and making the materials accessible to students. (See Mahler, 1976 for a more thorough discussion of implementation.)

Since the interviews for this study were conducted, continued contact with CERL has led to the observation of increased implementation activities. In all cases, original developers were initially involved in the maintenance and dissemination activities. More recently, new personnel have been added to assist with such efforts and, in a few cases, have assumed primary responsibility for implementation when the original developers have moved on to new projects. While implementation has functioned quite separately from the previous stages in many ways, it has been easier when it was planned from the beginning rather than added to the development of courseware designed for use only by the developers.

Controlling progress. It became necessary in nine of the thirteen production-oriented groups for a mechanism to be established for keeping courseware moving through the various stages. The major bottleneck was usually at the production stage because there were many loops which fed back to further refinement and because some developers became so engrossed in lessons that

they were never satisfied with their quality. Another bottleneck was caused by having to schedule "outsiders" for reviewing or student testing. Finally, documentation, maintenance, and dissemination were not particularly appealing activities for many courseware developers and were sometimes postponed indefinitely in favor of developing new lessons.

Two mechanisms were used to insure the orderly advancement of lessons through the various stages. The first approach was to have lesson developers fill out a form as they completed each step. This reporting kept them continually aware of the remaining steps and the current status of each lesson. The second approach was to designate a particular person, either the group leader or the group evaluator, to monitor the progress of each lesson and intervene when a problem arose. Such persons operated either formally with various forms and scheduled meetings or informally through persuasion and personal contact. Each of these procedures added significantly to the amount of administration needed by the group.

Such procedures were needed mostly by production-oriented groups with large and sometimes complex organizations. They were usually used only when externally imposed deadlines from the sponsoring organization needed to be met. Another contributing factor was that very few developers had much previous experience in curriculum development or computer applications.

Because they were learning how to use a new medium and how to develop courseware, it was difficult to estimate the time required for completion of each stage. Since the PLATO system was adding new hardware and software capabilities during this same period, it would have been impossible for people to be thoroughly familiar with this new medium. In fact, some of the courseware development groups had to be deeply involved in the hardware and software developments which were needed for their courseware. These topics are discussed more thoroughly in a later section on the influences of the PLATO system.

Comparison to other education R & D efforts. The primary emphasis of the PLATO project has been to develop a flexible, education-oriented hardware/software system rather than to develop curricular materials and a medium to deliver them. The leaders of this effort have been engineers and physical scientists, as well as other educators. Easley (1968) suggested, as an ideal, a team-approach to courseware development similar to the general instructional development system previously discussed. As more capabilities were added, dissatisfaction with a "systems approach" grew (Avner, 1975). In an effort to maintain maximum flexibility and to explore all possible uses of this relatively new medium, developers were encouraged to use their creativity individually. By the early days of PLATO IV in 1971-73, the idea had developed that all

college-level instructors could write lessons for use in their own courses, presumably eliminating the need for validation, documentation, or dissemination of courseware. This belief is still espoused in an official PLATO publication (Wood, 1975).

The great influx of funding in the early 1970's caused two problems for this approach. First, additional deadlines and independent evaluations were required by the funding agencies. Therefore, wasted efforts caused by trying out new ideas were less tolerable. Secondly, many more people were enticed into trying to develop courseware by full-time positions, release from teaching duties, the potential for royalties or job advancement, the glamour of a nationally recognized project, and the personal encouragement of the project staff. Without the previous self-selection hurdle of strong personal dedication in order to withstand the uncertainties and problems of working with a developing system, many of these people proved to be unproductive (Gjerde, 1973; House, 1974). For the past few years, there has been a shift back toward the more structured, team-oriented groups. Personnel and group structures will be discussed later in this paper; the point here is that the final stages of courseware development are once again receiving greater emphasis. It should be clear that there are advocates of both structured and unstructured approaches to PLATO courseware development. The ultimate value of each approach cannot yet be judged.

Since most of the literature on instructional development is prescriptive, that is, telling what should be done, it is not surprising that most of the models proposed are quite formal, elaborate, and rigid. The few papers which are descriptive of what has actually happened (Cashell, Lent, & Richardson, 1975) tend toward more flexible approaches. When the details of some rather structured models are examined (Reed, Ertel, & Collart, 1974) or when the people who actually produce the instructional materials are interviewed, it becomes apparent that many ad hoc decisions must be made and revised in most cases, and an iterative approach which includes modification of objectives and strategies is the rule rather than the exception. While PLATO's history of courseware development has varying emphases and many variations in procedures still exist, the overall effect does not seem to be very different from other educational R & D efforts.

Types of lessons.

Several different classification schemes for describing computer-based courseware have been proposed (Zinn, 1967; Grubb, 1972; Cody, 1973; Milner and Wildberger, 1974; Wang, 1976). They contain between five and thirteen categories, which are usually placed on a continuum of "increasing student control" (Milner and Wildberger, 1974) or "a progressive shift in the locus of control" from the designer to the student (Grubb, 1972).

Wang's Index to Computer Based Learning (1976) contains descriptions of 336 PLATO lessons or groups of lessons. Each entry is supposed to include the instructional strategy used, but unfortunately 225 of the PLATO lesson descriptions simply say that they used a "mixed" instructional strategy. The remainder used the following descriptors: drill and practice (56), tutorial (37), simulation (12), computer-managed instruction (11), problem solving (7), inquiry (6), gaming (6), diagnosis and prescription (1), socratic dialogue (1), testing (1), and intrinsic (1). Unfortunately, many of these terms are used by various groups in quite different fashions, and Wang does not provide explicit definitions. The total number of descriptors comes to more than 336 because several entries used more than one descriptor. With the large number of entries using the non-descriptive "mixed" descriptor and no information about how the 336 entries were selected, no conclusions about the instructional strategies of the 4000 hours of PLATO instruction claimed by Lyman (1975) and Wood (1975) can be reached.

There are presently no complete data on the relationship between lesson types and work-hours, personnel needs, or any other courseware development procedures. In a preliminary study, Avner (personal communication) found that first-year developers on the PLATO III system took more time to develop tutorial lessons than drill lessons, possibly because the former simply use more code.

Scope of the courseware.

The best way to organize a courseware development group, as well as the amount of work, is greatly influenced by the depth and breadth of the courseware to be produced. The scope needs to be measured along at least three dimensions. The first and most obvious is whether the materials will be used in an entire course, for a single unit within a course, throughout the entire curriculum of a department or school, or to some other degree of breadth. Secondly, there needs to be a decision regarding the depth or degree to which the computer-based materials are expected to teach the content. They may be intended for use as supplementary materials for those who want enrichment outside of class, as remedial work for those who cannot keep up with the regular class, as drill or practice on concepts taught in other ways, as the primary source of instruction for selected concepts or activities (such as simulated labs), or as the sole source of instruction for students at remote locations. The terms "adjunctive vs. mainline" (Bunderson, 1973) have been used in reference to this dimension, but it is at least a continuum if not a multi-dimensional concept. The third dimension is highly confounded by the first two and refers to the degree to which the individual parts of the courseware are expected to be related and coordinated. At one extreme, if supplementary treatment of single concepts is planned, coordination could be extensive but will probably be minimal. If,

however, primary instruction throughout an entire curriculum is planned, the individual lessons should be closely related to one another but could exist as a "smorgasbord" of possibilities to be selected by the individual instructor or student.

As the scope of the courseware increased, not only did the amount of coordination of the work increase, but there was also a greater need for planning and review.

Content areas.

It is not clear at this time that any broad content areas are not suitable for computer-based instruction; rather, it appears that certain types of instruction within each area are better suited for this medium than are others. Since PLATO lessons have been developed in more than fifty content areas, ranging from the kindergarten level to graduate courses, no area of education can or should be excluded at this time. Merrill (1975) has suggested some criteria for deciding if a particular instructional sequence should be programmed on PLATO, but there is certainly room for appropriate topics in all major content areas and age levels and for expansion of these criteria as new features are created. One conclusion that can be stated is that courseware production is easiest when the objectives can be clearly stated, and this process may be easier in some areas than it is in others. However, ease of development must be weighed against potential benefits and the need to explore new areas.

ORGANIZATION AND MANAGEMENT

Changing emphases for group activities.

All of the groups in this study shifted the emphasis of their work activities over the course of time. The first emphasis was characterized by exploration of PLATO's potentials. Some planned for this type of activity while others were forced into it because of false starts and changes in goals. While most interest was on the unique features of the system, there was also interest in the appropriate instructional strategies. This first emphasis lasted from a month to more than a year. The second emphasis was on developing instructional lessons, either to fit into a grand scheme or as independent modules determined by the interests of the various individuals. These lessons reflected the possibilities explored earlier and used material developed during that time, but the emphasis shifted to producing, reviewing, and revising lessons until they were useable. The third emphasis was an attempt to fit all of the lessons together into the on-going curricula of one or more regular classrooms and to validate their use. This effort often necessitated modifications, the development of a routing system, and continual liaison with the classroom instructors, particularly if such activities were not previously included. The fourth and final emphasis is characterized by the maintenance of the courseware and attempts to spread its use to other classrooms. At the time the interviews for this study were

conducted, no groups had reached this steady state. Since that time, almost all groups have moved into this implementation stage, and most have found that the developers have left and that this work must be carried out by staff members hired specifically for this work.

Models of group structure.

Each of the 16 groups had its own unique organization and history. To describe each of these would be tedious and confusing. Therefore, this discussion begins with four abstract models of group structure, which were based on an analysis of the 16 groups, and then compares the groups with these models.

The independent developer. Sometimes one person took responsibility for the development of instructional lessons. Such a person often sought advice but made the final decisions and did most of the work. They often shared ideas and critiques with other developers, but there was no direct coordination, organized group effort, or formal relationship in most cases. Independent developers needed expertise in a content area, instructional design, and TUTOR programming and have been called PLATO "authors." It was hoped at one time that many college faculty members would become independent developers (House, 1974; Wood, 1975).

The colleagueship. Sometimes several people of essentially equal status worked together with a commitment to cooperation and group decision-making. In most cases, specialization occurred

or was planned. In one group, one person took responsibility for administrative duties, another became an authority on intricate displays, and another specialized in observing student behavior and evaluating. However, all members retained responsibility and interest in the total effort, and no clearly defined hierarchy emerged.

The lesson designer with programming assistants. Subject matter experts, either faculty members or full-time developers, sometimes specified the content objectives and instructional strategy and then let a programming assistant design the actual lesson. These programming assistants, who were often students, worked with varying degrees of autonomy depending upon their own content and programming expertise and the desired involvement of the lesson designer. Contact between the two ranged from interactions on a daily basis to only formal reviews of completed work at intervals of a month or more. Some lesson designers would begin the actual programming and then let an assistant finish it, while others, who were not knowledgeable about the PLATO system or computer-based education in general, had to rely much more heavily on their programming assistants.

The support staff. Several groups added specialists in TUTOR programming, instructional design and evaluation, or audio-visual production. The TUTOR experts consulted with lesson designers and programming assistants, worked on particularly

complex routines, especially support routines such as "routers" and "drivers," and trained new group members to use the TUTOR language. The instructional design and evaluation specialists consulted with lesson designers and were often responsible for reviews and validation. The audio-visual production specialists were used when there was a major commitment to the use of microfiche or audio messages.

Comparison of models and groups. The correspondence between the models just presented and the groups which were interviewed for this study is shown in Table 3. The groups are listed according to the dates when they first began work on PLATO. The various phases were used to indicate different times at which groups changed their modes of operation. The length of each phase varied with each group.

Insert Table 3 about here

Some of the "groups" were not actually organized structures, particularly in their first phases. All of the non-military groups started as several separate efforts, each often exhibiting the characteristics of different models. Sometimes these efforts were located in different departments or even different institutions and had very limited contact with each other. Formal group structures and coordination activities came only when outside

funding permitted release from teaching duties or full-time staff members. The military sites all began with such funding.

Three of the groups were studied because they provided support for other courseware development efforts. Although the people in each of these groups did develop instructional lessons, often as training materials or on-line aids for other developers, their primary missions were not to develop lessons for student use. When they did produce lessons, either individual members worked closely with another group or the group functioned as a colleagueship to produce support routines and training materials. Because of this unique status, they are listed in Table 3 as pure forms of the support staff and are not included in most of the following discussions, which focus on the thirteen "production-oriented" groups charged with developing lessons for student use.

All but one of the production-oriented groups exhibited the characteristics of two or more models. The lone exception made such heavy use of the Military Training Centers group for support that it cannot completely be considered as a pure example of a colleagueship. All of the non-military groups without support staffs, and many with their own support staffs, made use of the PLATO Services Organization and the PLATO Educational Evaluation and Research group.

It should be noted that four of the first five groups and four of the last five groups have only one phase, while the groups in the middle of the table have more complex patterns. There appear to be several reasons for these characteristics. The four early groups with only one phase were all based upon substantial involvement by University of Illinois faculty members who demonstrated great interest and commitment early in the history of PLATO. They seemed to have known what they wanted to accomplish and how they wanted to work more clearly than developers hired for special projects. Also because of this base and the early starts, these groups were under less scrutiny than the remaining groups, which were all funded by NSF or ARPA. Therefore, some of their early phases or more subtle changes may have been missed. On the other hand, the latest one-phase groups were all either support groups or military groups and have relatively short histories. Not only may they have benefited from the experiences of earlier groups, but their non-academic bases had special consequences. Thus, we cannot be sure whether the lack of any independent developers in the last seven groups is the result of disenchantment with that mode of operation or is because military sites mitigate against such individualism.

Finally, most of the groups with convoluted histories are part of the National Science Foundation project and thus were greatly influenced by a second infusion of funds and accompanying

change in direction, which came after the project had been going for 18 months. In some cases, there was nearly a complete change in personnel from one phase to another. Since these groups had to be concerned with externally determined deadlines and with implementation outside of the university, there may have been more pressure to reorganize at various times. Also, since four of these groups were based at CERL, they were observed personally and more closely throughout their existences.

The success of independent developers was very mixed. Some of the most productive PLATO developers worked independently, but some others who tried never completed any useable lessons after as much as a year of full-time support. Many other independent developers were able to complete only a few trivial lessons. A great problem has been the identification of people with the necessary expertise in a content area, instructional design, and the TUTOR language along with dedication and creativity (see Mahler, 1976). Most of the successful independent developers learned the TUTOR language as it was being developed over several years. The newcomers, more than half of whom had little or no computer programming experience, either were overwhelmed or were limited to a small subset of the language, resulting in simple lessons. Another problem was that independent developers did not always use reviews, and factual errors, inferior instructional strategies, and poor programming were sometimes left in lessons

made available for general use. Without any coordination, there were examples of several independent developers who worked on lessons covering the same topics, and there was some tendency for independently developed lessons not to be used by other instructors. On the positive side, independent developers required little administrative overhead, and the more successful ones often worked for intrinsic rewards or professional recognition, rather than for full-time salaries. Also, some of these independent developers produced very creative and apparently effective lessons.

The colleagueships, while not usually requiring a great deal of formal administration or direction, did need a substantial amount of effort aimed at fostering cooperation and group spirit. Finding and keeping qualified people was also a problem. Since they were all of equal status, there was little chance of reducing costs by paying lower salaries for routine work. Because of the specialization that occurred in all colleagueships, there was less demand on each group member to provide expertise in all facets of courseware development. Lessons using sophisticated programming were possible without causing an undue burden on any individual to learn an unreasonable number of new skills. The sharing of plans and mutual reviews of lessons eliminated most of the factual errors, inferior instructional strategies, and

poor programming. The popularity of this group structure is evident in its being part of eleven of the thirteen production-oriented groups.

Lesson designers were mostly faculty members or full-time content experts with many responsibilities. Adding programming assistants seemed to make better use of these higher-paid, busy content experts. Several faculty members, who would not otherwise have done so, became involved with PLATO because they did not have to do their own programming. The success of these arrangements depended upon the programming abilities of the assistants, the amount of involvement of the lesson designers, and their ability to communicate with each other. Success was very limited when the programming assistant was unfamiliar with the content area and the lesson designer was unfamiliar with the PLATO system and computer-based education, in general. Not only was communication difficult in these circumstances, but creative interaction between the content specifications and the instructional design specifications was almost impossible.

Support staffs were either incorporated successfully with other group structures or were used extensively by other groups. TUTOR experts were the most common type of support staff members. Evaluation specialists, who usually had advanced degrees, were often primarily responsible for reviews and validation efforts. Some of the audio-visual production specialists had advanced

degrees and were responsible not only for the production of slides or audio discs but also for their design. Instructional design consultants were the least used type of support staff. It is unknown whether the additional expense of an internal support staff can be justified by greater productivity by the primary developers or greater quality in the courseware. Support staffs tended to be added as groups became larger and more complex, and it seems likely that sophisticated programs, validation, and peripheral devices would not have been used without the help of these specialists.

Influences of the PLATO System

A number of factors which influenced the organization and management of these groups were determined by the use of the PLATO system and were not under direct control of the individual groups. These factors often had differential effects on groups but operated mostly as a general context in which group decisions could be made.

Development of the PLATO System. The CERL policy of continuing development of the system and its language (Avner, 1975; Steinberg, 1975; Stifle, 1975; Tenczar, 1975) was advantageous in some ways but disadvantageous in others. It led to a system which is very flexible and well suited to the needs of a broad spectrum of educational applications. Author input had a significant impact upon the development of the TUTOR

language, as authors could request and usually receive new language featuers to facilitate their work. Of the experienced TUTOR programmers who were interviewed, none indicated ever being limited by the language. This continuing development has led to efforts in many areas and with many instructional techniques that could not have been imagined without ample experience with the medium. However, this expansion and refinement also created demands on programmers because it was coupled with instability. If someone did not use the system for a few months or weeks, many new or revised features often had to be learned before productive efforts could resume. Some people have spent considerable time in efforts only to find that a new system feature made the problem trivial. There have also been some people who found PLATO programming to be so confusing, demanding, and unstable that they decided that it was not worth the efforts required of them. As more PLATO systems are put into operation, continued development of the authoring language and system software will probably be restricted to one experimental system with the operationally oriented systems receiving updates at less frequent intervals.

Hardware and software limitations. Although the power of the PLATO hardware and software have created many opportunities for innovative educational use, several problems arose in the early period of courseware development which adversely affected

that process. These problems were largely created by an unanticipated competition for PLATO system resources. Initially, many groups, especially those remote from CERL, had difficulty obtaining terminals, often due to late delivery. As terminals became more widely available, authors found development hampered because of shortages of "lesson spaces"--allotments of disk storage. These shortages have generally been alleviated by acquisition of hardware. Eventually, critical difficulties were experienced with the allocation of Extended Core Storage (ECS). As more authors gained access to PLATO and as student usage increased sharply, it became clear that there was insufficient ECS to suport the resulting number of users. Although additional ECS was obtained, it became obvious that this resource had to be carefully managed.

Further restrictions were placed on authors because of competition for the "condensor," a program which translates the author's program into a format used by the PLATO computer. There was not enough computing power to allow authors to condense at will (sometimes several times a minute while debugging), so sharp curtailment of "condensing" was enforced.

In all cases, students were given higher priority in access to these features than authors, so that much on-line work during the daylight hours, when students were using the system, became difficult and sometimes impossible. Developers were frequently forced to adapt to a nighttime work schedule.

Development of some aspects of the system lagged behind others. In particular, peripheral devices such as the touch panel, slide selector and audio device had not reached a stage of development comparable with other hardware on the system. The touch panels, though reliable, were produced slowly at first and were consequently not readily available. On the other hand, slide selectors were widely available, but microfiche for the slide selectors were difficult to produce. The production process was rather long (two weeks at minimum), and the quality was not consistent. The prototype audio device was both unreliable and in short supply. Audio recordings were difficult to produce, and the end result was frequently undesirable. In operation, the device was often balky.

As a consequence of these problems, those groups which counted heavily on the operation of these devices in their courseware development were required to devote time and resources to maintenance and repair of equipment and feedback to design engineers. In some cases, it was necessary to redesign and reprogram lessons to minimize dependence on the peripheral devices. Subsequent improvements to the hardware and production processes have reduced the early problems with touch, slide, and audio so that future developers should not encounter the same difficulties.

Proximity to CERL. Having a group based at CERL had several advantages and a few disadvantages. First of all, users in Champaign-Urbana needed to be far less concerned with routine hardware maintenance and reliability. Many experienced technicians were close at hand, and there were fewer problems caused by telephone transmissions and delivery of equipment, especially during the early days of the project. Secondly, experienced PLATO programmers and lesson designers were readily available for face-to-face consultation, while remote users had to depend more heavily on on-line consultants located at CERL. Thirdly, this pool of experienced people located at CERL could often be drawn upon for staffing a new project.

Sources of advice. Most of the advice which was sought concerned how to program a particular sequence. Although some people first tried to get the information they needed from the on-line documentation called AIDS, most went to a fellow programmer or the TUTOR expert in their group. As the AIDS materials have been improved, they have been used more--even as the initial source of information. When a local resource person was not available or could not help, they contacted a member of the PLATO Services Organization or some other TUTOR authority at CERL either personally or through the on-line consulting capability, which allows both the consultant and the client to look at the appropriate part of the lesson while typing messages to each other across the bottom of the screen.

Since most developers had already decided on the content and instructional strategy of the lesson when they began production, they usually did not seek advice in these areas. Advice on these aspects sometimes came during training or the planning stage of courseware development but was only actively sought during the review stage.

Use of support routines. The parts of the courseware which do not contain the content-oriented instructional materials but which serve in the development or presentation of such materials are referred to as support routines. These routines may be integrated into the instructional lessons or may exist separately, using or being used by the instructional lessons as needed. At least three types of support routines are in use. "Drivers" are routines which set up the basic format of a type of instruction and into which different content can be placed. For example, some "drivers" provide the basic structure for multiple-choice questions, including placement on the screen, random ordering of alternative responses, corrective feedback, and storage of responses. It is then possible for someone with little programming experience to put in questions with appropriate response alternatives and designation of the correct answers. A large variety of these "drivers" have been written and made available for general use by their developers. A second type of support routine is for the collection and manipulation of student data. Some data can be collected automatically by system-provided routines. This type of data is intentionally rather general, with more focused data collection being left to the individual

developer according to his own specific needs. In addition to the collection data, it was also necessary to provide programs for data summary and analysis (Tatsuoka and Siegel, 1975). The third type of support routine is the "router" which routes or transfers students to the instructional material which they should see next. A system-provided router has been developed, but some groups used elaborate routers of their own design. A heavy commitment to the use of support routines had several implications for courseware production. First of all, they usually required the skills of an experienced fulltime programmer to develop. Secondly, such routines required a substantial commitment of time and resources which were justified only if used extensively. For example, one group claimed to have devoted approximately 1,000 hours to the development of a "driver" which now allows them to add an hour of instructional time in one hour of work. If only one hour of such instruction is used, it would have required 1,001 work hours, but if 1,000 hours of such instruction are used, each will have required only two work hours. Since it is doubtful that 1,000 hours of one format would be used, the actual figure should fall somewhere in between the extremes.

As can be seen in the above example, some groups have written elaborate support routines which were intended for general or repeated use, while other groups have used "quick and dirty" methods with no generalizability. Several of these efforts

were later made obsolete by new system features, such as the system-provided router. In the future, support routines should require less time and effort because of the availability of existing routines, but some unique situations, requiring new routines or adaptations of existing routines, are likely. Also, the application of existing routines to the particular courseware could demand the attention of a programming specialist.

Characteristics of groups.

In addition to the models of group structure, groups differed in their methods of selecting members of the group and the training received by the new members. In many cases, these characteristics came about as the result of gradual development rather than on the basis of overt decisions.

Selection methods. When a project relied upon subject matter specialists of professional status, especially faculty members, to volunteer for involvement in courseware production, the only selection possible was self-selection. If there were no restrictions on the length of time, the computer availability, or number of the support personnel, such a procedure allowed the most motivated and capable developers to emerge. However, there was a tremendous risk of wasted effort. Some incapable persons continued to work because of the glamour of the medium, a greater chance for advancement, or a variety of personal reasons. Some capable persons discontinued their efforts because of difficulties

stemming from competitions for limited resources, desires to use their time in the most profitable ways, or personal reasons. The self-selection approach must account for these failures, as well as the successful developers, when determining costs and efficiency.

When the leaders of a courseware development group were faced with deadlines or limited resources, as was usually the case, they had to make choices among available personnel. When the lesson designers were hired on a full-time basis, released from teaching or other duties, paid as consultants, or even just given access to resources, some type of overt selection also took place. Since none of these jobs were very routine or well defined, the first criterion was interest or motivation. Relevant backgrounds are discussed in a later section, but beyond these specific skills and experiences, personal characteristics such as flexibility, prolonged dedication to a task, and interest in innovation were considered.

At the time of this study it was necessary to observe each person working on the PLATO system in order to determine the eventual level of productivity. Consequently, many groups had to readjust personnel commitments when people who were hired did not prove to be productive. One of the people most heavily involved in the training of new lesson designers and programmers claimed to be able to predict future programming ability with 90%

accuracy after the first day of training but not on the basis of previous, documented background. If such predictions are possible, it should also be possible to devise a task-specific test which could aid in these judgments before training begins (Popham, et al., 1974).

TUTOR training methods. The one area of competency for which specific training programs were developed was knowledge of the TUTOR language. Such training also included PLATO lesson design, which is a part of instructional design. During the early days of PLATO IV and throughout the life of PLATO III, such training was very informal. Evidently, most people who expressed interest were given a demonstration of some existing lessons, shown how to "sign on," given a lesson space with which to work, and told to ask questions whenever they needed help. Such people went through whichever existing lessons they heard about (as a student and, perhaps, looking at the TUTOR code), reviewed the scant documentation that existed online, and asked whoever was sitting at the next terminal if a problem arose. They soon learned who were the most knowledgeable people, including the systems programming staff, when they got to more complex questions. This informality was not only feasible because of the close proximity of all PLATO users, but it was also necessary because the language was undergoing almost daily revision, often

in response to the questions of the users. Several factors such as the increasing stability of the TUTOR language, more people at remote locations, and the organization of formal courseware development teams led to the feasibility and need for more formal training methods.

Three major types of training evolved. First, accredited courses which included training in the basics of TUTOR, as well as general discussions of computer applications in education and instructional development methods or an analysis of PLATO as a computer system, were offered by several departments of the University of Illinois, including extension courses in Chicago. The courses used a workbook (Ghesquiere, Davis, and Thompson, 1974) and its accompanying PLATO lessons.

A second effort was aimed mostly at people who accepted full-time jobs to develop PLATO materials, especially in the military. Most of these people participated in one- or two-week intensive workshops at the University of Illinois. During the workshops, the people learned the basics of the TUTOR language and, if necessary, the rudiments of lesson design and validation. The workshops used some lessons and a manual by Bohn (1973). These basic introductions to TUTOR were estimated to take 20 - 80 hours of work, depending upon the person's background. People with computer programming, instructional development, and mathematical backgrounds tended to take less time although such backgrounds were not necessary for eventual success.

Finally, advanced training in the TUTOR language, following either of the two previous alternatives, was much more informal and was closely related to the person's actual work. When problems arose, assistance was sought from various publications, such as Sherwood (1974) and E. Avner (1975, 1975b), from the online documentation available in lesson AIDS and elsewhere, and from various consultants. CERL set up the PLATO Services Organization to provide consultation either through personal contact or through the various communication featuers of the PLATO system. In addition, many groups found it convenient and, perhaps, necessary to hire or develop a local TUTOR expert who could provide consultation and advanced training. The comprehensiveness of a person's knowledge of TUTOR varied greatly and depended upon both the person's background and the types of lessons being written. It took anywhere from a few weeks to a year before the lessons being written were of sufficient quality to warrant keeping and using them. When this training was for an individual who was joining an existing group, the learning often took place while the newcomer was working productively under the direction of a competent group member. The amoung of direction usually decreased as the degree of the newcomer's competency increased. When a new group was being formed, there was more pressure to advance beyond the learning stage but also less chance for

competent supervision of individuals. Several groups, therefore, hired a few highly skilled TUTOR programmers to set up a formal training program.

Characteristics of individuals.

Within most groups, there were differences in the backgrounds sought for various positions, in the interests of people as they became more involved, in the amount of work expected, and in the rewards given for PLATO work. All of these factors can be considered as characteristics of individuals.

Relevant backgrounds. The models of group structure described earlier call for persons with different backgrounds or, conversely, were differentially appropriate when different backgrounds were found in the development team. In some cases, the model came first; in other cases, the people came first; in most cases, one or two people began the effort, decided upon a particular working relationship, and then hired people with appropriate backgrounds to fill in the rest of the team. Regardless of the particular model or combination of models, the following areas of expertise (or, at least, responsibility) had to be included: subject matter content, instructional development including knowledge of appropriate uses of computer-based systems, and TUTOR programming. In addition, expertise in the production of audio/visual materials was necessary when audio messages or slides were used. When the materials were intended for general

use, someone had to be responsible for monitoring student performance, soliciting expert or peer reviews, and validation.

Because the relationship between groups and models of group structure is so complex, it is not possible to give the relevant backgrounds of people according to the models. Therefore, Table 4 lists the relevant backgrounds according to the criteria used in orginally selecting the groups. The first four categories of experience indicate levels of formal education, and the last three categories indicate actual working experience prior to initial involvement with the PLATO project. All interviewees were asked about their backgrounds (see appendix), but not all people in all groups were interviewed. Therefore, while these data are generally indicative of relevant backgrounds, specific numbers, particularly the smaller ones, may be slightly inaccurate.

As indicated in Table 4, projects at colleges and universities were staffed mostly by people with advanced degrees in computer-related fields, educational fields, and a variety of other content areas. Almost all of the advanced degree holders working for military projects were at one base, where materials for a para-medical program were being developed. With regard to working experience, approximately 61% of the people had prior teaching experience. The slightly lower 50% for the university

groups was due to the involvement of students working under the direction of faculty members. The extremely small number of people with prior experience in developing other instructional materials is quite remarkable. It is impossible to say whether this phenomenon was advantageous or disadvantageous to PLATO courseware development efforts. It can also not be determined whether the unique characteristics of PLATO courseware development procedures were the cause or the effect of the lack of curriculum development experience. Finally, the fact that 68% of the developers had little or no previous computer experience seems to be in keeping with the idea of developing a computer-based educational system that can be used by many people. However, it is also clear that, even though the system developers are not included in these figures, a significant number of developers did (and perhaps had to) have substantial computer-related backgrounds.

Insert Table 4 about here

Levels of individual interest. In addition to learning the TUTOR language and any other necessary competencies, there appeared to be several levels of interest through which PLATO courseware developers proceeded. In the first level, the person learned the basics of TUTOR and the PLATO system. In the second level, the interest centered on exploring the TUTOR language and

the capabilities of PLATO. There seemed to be a fascination with the hardware and software, and some people, particularly those without experience as instructors, spun-off to become TUTOR experts or computer system programmers and developers. Others without instructional experience remained at this level and worked as TUTOR programmers or coders under close supervision. The third level was characterized by an interest in the design of individual PLATO lessons in order to make them instructionally sound rather than simply interesting things to program. If the person was actively teaching at the time, there was often an attempt to try various ideas in the classroom. People at this level can be called lesson designers. Some people went back and forth between the second and third levels as new instructional ideas necessitated better programming skills and better programming skills generated new instructional ideas. A similar pattern was found by Avner (personal communication) in an unpublished study of 27 first-year PLATO developers. Those people who went into the fourth level began to look beyond individual lessons and think about sequencing, routing, the relationship of lessons to a curriculum, the use of lessons in various classroom contexts, and the principles of learning which underlie the materials. Finally, the few people who entered the fifth level became interested in the design of instructional systems. They developed opinions and ideas not only about the

design of the instructional materials but also about the process of developing such materials and the organization of appropriate development teams. Some people with prior experience in curriculum development jumped from the first level, to the fourth or fifth level, and a few of these later went back to learn the specific skills of the second and, possibly, third levels.

Full-time vs. part-time work. The circumstances under which a group was formed largely determined whether the members worked full-time or part-time. When a new group was brought together for the specific purpose of developing courseware for a particular project, the members usually worked full-time. Students who worked as programming assistants or support specialists worked part-time. The real question arose with regard to the role of faculty members. Some groups found that professionals were more productive when they were not distracted by other responsibilities. On the other hand, continued contact with students of the target population sometimes stimulated ideas, tempered them with reality, and provided a useful forum for trying out new approaches. The role of the professional was important. If the person was to be only an initiator and/or reviewer, part-time work was much more feasible and probably preferable than if actual production was also part of the responsibilities.

When a project had the complete freedom to determine its personnel composition, the goals of the project determined the desirability of full-time or part-time workers. When the major or sole goal was the production of appropriate courseware, full-time concentration by professionals, programmers, and specialists was needed. When the education of students in the use of computers for instruction was a goal, part-time programmers and specialists could be learning while doing. When commitment to use of the system by faculty members was a goal, wider involvement by part-time professionals was used. A final consideration was that when a project was just beginning, it was easier to terminate an unproductive part-timer than an unproductive full-time employee.

Rewards for PLATO work. The reward structure of the sponsoring institution determined the extrinsic rewards for PLATO work. In the military and projects with external funding at academic institutions, there were usually full-time jobs for the duration of the development effort. Academic institutions also released faculty members from other duties in order to develop courseware or set up a formal policy to equate such efforts with teaching, research, and publications in the determination of promotion and tenure. Unfortunately, when the regular procedures, usually involving a committee review, have been left intact, courseware development has often not been equated with research

or other scholarly work. There are also plans to pay royalties to the developers of lessons, based on the amount of use. Of course, if the developers were paid for their work, the royalties may go to the sponsoring institution.

Many of the people developing PLATO courseware were motivated, in addition to the extrinsic rewards described above, by rewards which were intrinsic to the work and intangible. Some people simply enjoyed the challenge of programming a computer to carry out a specific task; some liked the environment and spirit of a developmental effort; some believed that their efforts will have a significant, positive effect on their students or education in general. There were even high school and undergraduate students who worked as programming assistants in order to get lesson spaces in which they could design their own materials. These intrinsic rewards seemed to drive many PLATO developers to work much more than the normal work week. Such unpaid overtime work makes it difficult to assess the costs of development. It should not be assumed that the intrinsic rewards were sufficient motivation for most people. With the possible exception of uninformed lesson designers providing minimal supervision for highly competent programming assistants, the time investment for this work was too great to expect anyone to do a satisfactory job without released time or regular pay.

Costs of courseware development.

As indicated in the appendix to this paper, the interviews did include attempts to get estimates of the number of work-hours spent and the amount of material developed. Most interviewees, however, were unable to give anything but gross estimates. The estimates that were received ranged from ten to a thousand work-hours for single lessons. Some groups had previously determined an estimated range by some unexplained means, and all group members quoted the same range. Since these figures were undocumented, there is no way of knowing what activities and cost factors were included.

Grimes (1975) has recently reported on the "cost of initial development of PLATO instruction in veterinary medicine." His estimated average cost of $828.00 per instructional hour, or $1.93 per student-contact hour, does not include expenditures for "computer usage and salaries of most instructors for and with whom lessons were developed . . . except those of released-time personnel or those of instructors paid by the PLATO Project." He did include the cost of equipment, including PLATO terminals. The scope of the courseware is indicated as "more than 50 lessons" with "approximately 317 instructional hours" using simulations, games, problem-solving programs, and interactive tutorials which are interspersed throughout the four-year veterinary medicine curriculum. The entire project, including the start-up period and an unspecified validation effort, had lasted four years at

the time the report was written. Grimes concludes with the statement that the PLATO effort at the "University of Illinois College of Veterinary Medicine has been developmental. The expense of continuing the project at this college or initializing a similar project at another college should be much less as a result of this experience."

Dare, et al. (1975) of the Aberdeen group reported on the costs of development of PLATO courseware for a machinists' course and for a course on the construction and interpretation of tests. This accounting includes charges for the on-line and off-line time spent by members of the project staff (including administrators) for the period from July 1973 to March 1974 during which approximately 30 instructional hours of courseware were developed. However, it does not include charges for the assistance provided by the Military Training Centers group at CERL. Costs for terminals, communications, and computer usage are presented separately. The average development time per instructional hour was reported as 283.6 hours. This development time was costed at $8.00 per hour, a figure reported to be the actual average hourly salary of the project staff, giving an average development cost of $2,268.80. Dare, et al. did not report estimates of the costs of development per student contact hour nor did they project the number of students to be trained

annually with these materials. Assuming 200 students per year would use these materials for five years, the average cost of courseware development would be $2.27 per student-contact hour.

SUMMARY AND RECOMMENDATIONS FOR FURTHER RESEARCH

Since this study was broad and preliminary in nature, it raises more questions than it answers. Hopefully, it will provide a starting point for further research efforts. In an effort to promote and focus such investigations, this final section highlights the important finding, suggests a number of hypotheses and topics in need of research, and discusses appropriate methodology.

Hypotheses and Topics

The following summary and suggestions coincide with the outline of the main body of this paper. Interested readers are referred to appropriate preceeding sections for background information.

I. Courseware

A. Stages of courseware development

In order to provide a comprehensible description of the variety of procedures used to develop PLATO courseware, five stages (planning, production, review, validation and documentation, and implementation) were proposed. A description of their components and interactions as found in the

courseware development groups under study is included. The primary questions concern the generalizability of this model as a descriptive device and its usefulness in planning future efforts. Other research is needed to discover the best order in which planning specifications and reviews should be made in various situations, the personnel who should be responsible for each stage, and the interactive effects of each stage on total production time. At this time, we conclude that all stages should be anticipated and included in every courseware development effort.

B. Types of lessons

Several schemes for classifying types of lessons are discussed. Once one has been accepted and lessons have been classified, the hypothesis that more complex and interactive lessons require more total development time but are more effective for certain educational goals should be tested. The differential effects of complex lessons on the various stages of courseware development should also be investigated.

C. Scope of the courseware

The proper conditions for various levels of the scope dimensions (breadth, depth, and degree to which individual lessons are coordinated) need to be specified, and the effects of these levels on group structures, total development time, and the stages of courseware development need to be studied.

D. Content areas

More information on the suitability of content areas and topics within content areas for computer-based education is needed.

II. Organization and management

A. Changing emphases of group activities

The first question is whether or not the four emphases discussed (exploration, lesson production, curriculum integration, and implementation) are realistic, and the second is how progress to the final emphases can be encouraged.

B. Models of group structures

The four proposed models are the independent developer, the colleagueship, the lesson designer with programming assistants, and the support

staff. Research questions include the appropriateness and productivity of each model in different situations, their advantages and problems, and the effect they have on the eventual courseware and the stages of courseware development.

C. Influences of the PLATO system

To research this area, a comparison with other computer-based systems and other instructional delivery systems should be made. Within the PLATO system, topics for investigation include the advantages, disadvantages, and determinants of the following topics: system development occurring simultaneously with courseware development, the usefulness of peripheral devices (slides, audio, and touch panel), types and sources of advice and information, and the usefulness of support routines.

D. Characteristics of groups

The personnel selection and training methods used by the groups in this study were described. Pre-selection measures of courseware development abilities should be developed, and their reliability and validity should be determined.

The hypothesis that volunteers are more (or less) productive than full-time employees should be investigated. The factors which affect the length of training time should be clarified.

E. Characteristics of individuals

Relevant educational backgrounds and work experiences, as well as the levels of personal interest, were described but need to be clarified. For each stage of courseware development and each model of group structure, the advantages and disadvantages of full-time vs. part-time work and the rewards for courseware development work need to be determined.

F. Costs of courseware development

The personnel time and other costs associated with all of the above options should be determined. Also, the length of "calendar" time, as opposed to the number of work-hours, for each option is also needed. The products of these efforts should be properly measured. Presumably, such a measure would include the number of students who use the materials and the length of time per student, but it may also include the educational effectiveness of the materials and the

value of the other forms of instruction being replaced. The costs of PLATO courseware development should be compared to the development costs for other forms of instruction.

Methodology

Because this study attempted to delve into a relatively new area without many guides for substantive topics or methodology, the primary method used for collecting data had to be exploratory and flexible. Attempts at designing and distributing a questionnaire, looking at usage data recorded by the PLATO system, and asking people to keep a record of their activities for a week were relatively unsuccessful because the data were either very difficult to organize and interpret or they were obviously incomplete and distortions of the real procedures. Semi-structured interviews (see appendix) were finally selected because they could be used to examine past stages of development, even though the accuracy of recall of details is questionable under such conditions, and could be adapted to the individual interviewee's areas of knowledge. Another important consideration was that people who were unwilling to spend twenty minutes filling out a questionnaire were willing to spend one or more hours talking with a knowledgeable, interested colleague. Most of the interviews were recorded on audio tapes, but the primary records

used in writing this report were summaries written by the interviewers, using the form in the appendix. The form was designed after the first group had been interviewed and was also based on a great number of questions, observations, and hypotheses which had been generated by many people over the previous two years.

Many of the hypotheses and topics of the previous section could and should be studied independently, using a wide variety of questionnaires, tests, interviews, on-line measures, and other forms of data. However, the complex and interacting nature of many topics indicates the need for a more comprehensive study. In an effort to supplement interviews with more objective data, it is recommended that a mechanism for collecting data on time spent in various activities be made available to all PLATO courseware development groups and, so far as possible, to other courseware development efforts.

Insert Table 5 about here

A possible mechanism would be an interactive lesson or tailored questionnaire which asks each developer to indicate on a regular, perhaps weekly, basis how much time has been spent in each of several activities. Another possibility would be to have an independent observer use time-sampling techniques to gather

such data. The categories listed in Table 5 are a starting point but could be modified by each project to fit individual circumstances. For instance, some directors may want to list each lesson under development instead of the categories of instructional lessons indicated. Presumably, some categories, such as "audio messages" or "user recruitment," would be inappropriate for some people or at some times and could be eliminated under such circumstances. Equipment and miscellaneous costs could be filled in separately. There would also have to be accompanying descriptive information about the courseware development models being used, the selection procedures, the reward structure, the scope of the courseware, the number of instructional hours, the expected number of students who would use the courseware and the other topics raised in this paper. After a sufficient number of projects have used this data collection system to analyze their own developmental efforts, there could be an anonymous data base which could be used by new projects to predict the consequences of various decisions. Through such a process, courseware development procedures could be improved just as the courseware products of these efforts are improved. Hopefully, this paper is a step in that direction.

Footnotes

[1] Support for this study was provided by a contract given to Educational Testing Service (ETS) from the Advanced Research Projects Agency of the Department of Defense (Contract No. MDA 903-74-C-0290). The authors wish to thank Allen Avner, James McKenney, and Martin Siegel for their helpful comments during the course of this study but, of course, assume full responsibility for the contents of this paper. Requests for reprints should be sent to Dr. William A. Mahler, Office of Academic Affairs, University of Wisconsin, Oshkosh, WI 54901.

[2] At the time of this study, Dr. Mahler was an associate research psychologist for ETS. He is now a research associate in the faculty development program at the University of Wisconsin - Oshkosh.

[3] At the time of this study, Mr. Misselt was a graduate student in educational psychology at the University of Illinois at Urbana - Champaign and a research assistant for ETS. He is now an evaluator for the Military Training Centers project at the Computer-based Educational Research Laboratory, University of Illinois at Urbana - Champaign.

[4] At the time of this study, Mr. Schell was a graduate student in computer science at the University of Illinois at Urbana - Champaign and a research assistant for ETS. He has now returned to full-time graduate study.

[5] PLATO is an acronym for Programmed Logic for Automated Teaching Operations, a computer-based educational system developed at the University of Illinois at Urbana - Champaign.

REFERENCES

Alderman, Donald L. and Mahler, William A. The evaluation of PLATO and TICCIT: Educational analysis of the community college components. Princeton, NJ: Educational Testing Service, Project Report 73-49, 1973.

Alpert, Daniel and Jordan, Pauline R. Community college users' report: Fall, 1975. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1976.

Anderson, R. C. et al. Current research on instruction. Englewood Cliffs, NJ: Prentice-Hall, 1969.

Anastasio, Ernest J. and Morgan, Judith S. Factors inhibiting the use of computers in instruction. Princeton, NJ: EDUCOM, Interuniversity Communication Council, Inc., 1972.

Atkinson, Richard C. Ingredients for a theory of instruction. American Psychologist, 1972, 27, 921-931.

Atkinson, Richard C. and Paulson, J. A. An approach to the psychology of instruction. Psychological Bulletin, 1972, 78, 49-61.

Avner, Elaine. Summary of TUTOR commands and system variables. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, PLATO User's Memo No. 1, 1975.

Avner, Elaine. Basic bit operations. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, PLATO User's Memo No. 2, 1975.

Avner, R. Allen. Prototype reporting format for curriculum or lesson development. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, PLATO Evaluation, 1973.

Avner, R. Allen. CAI design as an evolutionary process. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Avner, R. Allen and Tenczar, Paul. The TUTOR manual. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, Report X-4, 1969.

Baker, Eva L. The technology of instructional development. In Travers, Robert M. W. (Ed.), Second handbook of research on teaching. Chicago: Rand McNally, 1973.

Bennett, J. A. Interactive computer-based lessons for engineering education. Engineering Education, 1975, 261-267.

Bohn, Robert. An introduction to basic elements of the TUTOR language. Urbana, IL: MTC Group, Computer-based Education Research Laboratory, University of Illinois, 1973.

Briggs, Leslie J. Handbook of procedures for the design of instruction. Pittsburgh, PA: American Institutes for Research, Monograph No. 4, 1970.

Broderick, William A. Instructional development for the Florida PLATO project: Process and evaluation. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Brunner, Jerome. Toward a theory of instruction. Cambridge, MA: Harvard University Press, 1966.

Bunderson, C. Victor. The TICCIT Project: Design strategy for educational innovation. Provo, UT: Institute for Computer Uses in Education, Brigham Young University, Technical Report, No. 4, 1973.

Carnegie Commission on Higher Education. The fourth revolution: Instructional technology in higher education. New York: McGraw-Hill, 1972.

Cashell, Jane G., Lent, Richard M., and Richardson, Penny L. Styles of instructional development. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Cody, Ronald. Computers in education: A review. Journal of College Science Teaching, 1973, 22-28.

Dare, Frank C., Hill, Craig E., Hall, F. A. and Wofford, B. A. Evaluation of the PLATO IV system in a military training environment. Aberdeen, MD: U. S. Army Ordinance Center and School, Aberdeen Proving Ground, 1975.

Diamond, Robert M., Eickmann, Paul E., Kelly, Edward F., Holloway, Robert E., Bickery, Tan R., and Pascarella, Ernest T. Instructional development for individualized learning in higher education. Englewood Cliffs, NJ: Educational Technology Publications, 1975.

Easely, J. A., Jr. A project to develop and evaluate a computerized system for instructional response analysis. SIRA Report to the U. S. Department of Health, Education, and Welfare, Office of Education, 1968.

Eisele, J. E. Systematic planning of curriculum and instruction. Educational Technology, 1973, 13, 9-12.

Eschenbrenner, J. A. and Lamos, J. P. The development and evaluation of the AIS courseware. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Essex, D. L. Lesson development and review. Urbana, IL: PLATO Project, School of Basic Medical Sciences, University of Illinois, 1975.

Felker, D. B. and Shettel, H. H. A methodology for evaluating a systems approach to instructional technology. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Fraley, L. E. and Vargas, E. A. Academic tradition and instructional technology. Journal of Higher Education, 1975, 46, 1-15.

Francis, Larry, Goldstein, Merle, and Sweeney, Eileen. Lesson review. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, PLATO User's Memo No. 3, 1975.

Gagné, Robert M. (Ed.). Psychological principles in system development. New York: Holt, Rinehart, and Winston, 1962.

Gagné, Robert M. The conditions of learning. New York: Holt, Rinehart, and Winston, 1970.

Gagné, Robert M. and Briggs, Leslie J. Principles of instructional design. New York: Holt, Rinehart, and Winston, 1974.

Gagné, Robert M. and Rohwer, W. D., Jr. Instructional psychology. Annual Review of Psychology, 1969, 20, 381-418.

Gerlach, V. S. and Ely, D. D. Teaching and media: A systematic approach. Englewood Cliffs, NJ: Prentice-Hall, 1971.

Ghesquiere, James, Davis, C., and Thompson, C. Introduction to TUTOR. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1974.

Gjerde, Craig L. Introducing a technological innovation into the community colleges. Unpublished dissertation, College of Education, University of Illinois at Urbana-Champaign, 1974.

Glaser, Robert. The design of instruction. In Goodlad, J. I. (Ed.), The changing American schools: 65th Yearbook of the National Society for the Study of Education. Chicago: University of Chicago Press, 1966.

Glaser, Robert and Resnick, L. B. Instructional psychology. Annual Review of Psychology, 1972, 23, 207-276.

Gow, D. T. and Yeager, J. L. The design and development of individualized curriculum materials for higher education. Journal of Higher Education, 1975, 46, 41-54.

Grimes, George J. Cost of initial development of PLATO instruction in veterinary medicine. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, Report X-43, 1975.

Grubb, Ralph E. A design language for computer-assisted instruction. Newburyport, MA: Entelek, 1972.

Hillelsohn, Michael J. Project IMPACT courseware subsystem: Volume I--Innovative procedures for development and administration. Alexandria, VA: Human Resources Research Organization, 1974.

House, Ernest R. The politics of educational innovation. Berkeley, CA: McCutchan Publishing Co., 1974.

Hunter, Beverly, Kastner, Carol S., Rubin, Martin L., and Seidel, Robert J. Learning alternatives in U. S. education: Where student and computer meet. Englewood Cliffs, NJ: Educational Technology Publications, 1975.

Johnson, S. R. and Johnson, R. B. Developing individualized instructional materials. Palo Alto, CA: Westinghouse Learning Press, 1971.

Kemp, J. E. Instructional design: A plan for unit and course development. Belmont, CA: Fearon Press, 1971.

Kozma, R. B., Ross, R., and Hitch, E. A systems approach to diagnostic-prescriptive instruction. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Locatis, Craig. Some uneasy inquiries into instructional development. Educational Technology, 1973, 13, 46-50.

Lyman, Elisabeth R. PLATO highlights. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1975.

Lyman, Elisabeth R. PLATO IV curricular materials. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1975.

Mahler, William A. The interaction of courseware development and implementation: Functions and models. Educational Technology, 1976, 16, 9-11.

McKeachie, Wilbert J. Instructional psychology. Annual Review of Psychology, 1974, 25.

McKeown, James C. PLATO instruction for elementary accounting. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, Report X-42, 1974.

McKeown, James C. and Lenehen, Thomas K. Educational effects of computer-based instruction in elementary accounting. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1974.

McManus, J. F. IDCMS: A computer-based system to facilitate educational product development. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Merrill, M. David. Instructional design: Readings. Englewood Cliffs, NJ: Prentice-Hall, 1971.

Merrill, M. David. Toward a theory-based approach to instructional development. Provo, UT: Brigham Young University, Working Paper No. 16, 1972.

Merrill, M. David and Boutwell, R. C. Instructional development: Methodology and research. In Kerlinger, F.N. (Ed.), Review of research in education. Itasca, IL: F.E. Peacock, 1973.

Merrill, Paul F. Training of personnel for the Florida PLATO projects. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Milner, Stuart and Wildberger, A. M. How should computers be used in learning? Journal of Computer-based Instruction, 1974, 1, 7-12.

Montanelli, Richard G. Computer science 103 PLATO experiment: Fall, 1974. Urbana, IL: Department of Computer Science, University of Illinois, 1975.

Penta, F. B. A systems model for the development of instructional materials. Educational Technology, 1973, 13, 12-16.

Popham, W. James. Preparing instructional projects: Four development principles. In Baker, R. and Schutz, R. G. (Eds.), Instructional product development. New York: Van Nostrand, 1971.

Popham, W. James, Cary, G. R., Chilstrom, B., Leps, A. L., Miller, T. M., and Saxe, R. Performance tests for instructional developers: An exploratory investigation. Educational Technology, 1974, 14, 33-34.

Reed, F. C., Ertel, P. Y., and Collart, M. E. A model for the development of computer assisted instruction programs. Educational Technology, 1974, 14, 12-20.

Schutz, R. G. The nature of educational development. Journal of Research and Development in Education, 1970, 3(2), 39-64.

Schutz, R. G. and Baker, R. An overview of product development. In Baker, R. and Schutz, R. G. (Eds.), Instructional product development. New York: Van Nostrand, 1971.

Sherwood, Bruce A. The TUTOR language. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1974.

Sherwood, Bruce A. and Stifle, Jack. The PLATO IV communications system. Urbana, IL: Computer-based Education Research Laboratory, Report X-44, 1975.

Smith, R. G. The design of instructional systems. Alexandria, VA: Human Resources Research Organization, 1966.

Smith, N. L. and Murray, S. L. The status of research on models of product development and evaluation. Educational Technology, 1975, 15, 13-17.

Steinberg, Esther R. The evolutionary development of CAI courseware. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Stifle, Jack. The PLATO IV student terminal. Proceedings of the Society for Information Display, 1972, 13, 35.

Stifle, Jack. The PLATO IV student terminal. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1974.

Stifle, Jack. The evolutionary development of CAI hardware. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Tatsuoka, Kumi and Siegel, Martin. Manual to accompany the "area" summary data and analysis package. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1975.

Tenczar, Paul. The evolutionary design of CAI software. Paper presented at the Annual Meeting of the American Educational Research Association, Washington, D. C., 1975.

Wallen, C. J. SATE: A systems approach to developing instructional programs. Educational Technology, 1973, 13, 27-32.

Wang, Anastasia. Index to computer-based learning. Milwaukee, WI: Instructional Media Laboratory, University of Wisconsin-Milwaukee, 1976.

Wood, Nancy. The PLATO system. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1975.

Yelon, S. L. An examination of systematic development of instruction for non-residential colleges. Educational Technology, 1973, 13, 36-44.

Zinn, Karl L. Computer technology for teaching and research on instruction. Review of Educational Research, 1967, 37, 618-634.

Table 1

Description of the Study Sample

	Groups	Dates of Interviews	No. of People Interviewed
NSF Project	Elementary Reading	June, 1974; March, 1975	8
	Elementary Mathematics	August-September, 1974	19
	Accounting	June, 1974; December, 1974	4
	Biology	December, 1974; March, 1975	4
	Chemistry	July, 1974; December, 1974; March, 1975	5
	Community College English	July, 1974; December, 1975	11
	Community College Mathematics	July, 1974; December, 1974	6
ARPA Project	Aberdeen Proving Grounds	December, 1974	6
	Chanute Air Force Base	January, 1975	9
	Sheppard Air Force Base	February, 1975	11

Table 1--Continued

	Groups	Dates of Interviews	No. of People Interviewed
Support Groups	Military Training Centers	August, 1974	9
	PLATO Services Organization	August, 1974	4
	PLATO Educational Evaluation and Research	August, 1974	4
University Groups	Basic Medical Sciences	February-March, 1975	9
	Foreign Languages	July, 1974	7
	Veterinary Medicine	August, 1974	6

Figure 1

Stages of Courseware Development

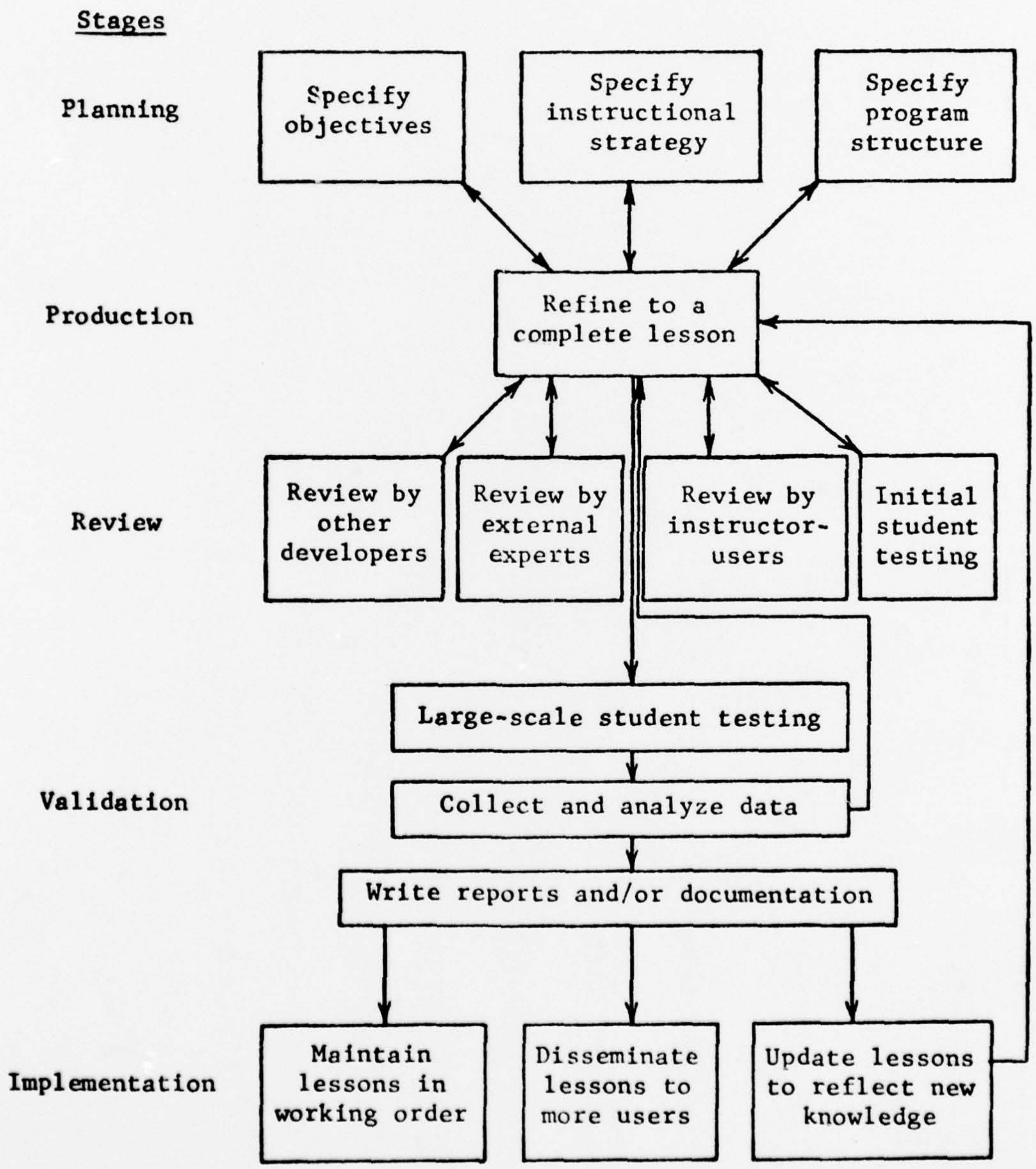

Table 2

Number of Groups Using Various Reviewers and Targets of Reviews

Targets of Review	Reviewers			
	Other Developers	External Experts	Instructor-Users	Student Testing
Content	12	3	7	8
Instructional Design	12	3	5	11
Coding	10	1	0	7

Table 3

Comparison of Models of Group Structure and PLATO Groups

Groups (With Starting Dates)	Models			
	Independent Developer	Colleagueship	Lesson Designer with Programming Assistants	Support Staff
Foreign Languages (Sept., 1966)	Phase 1	Phase 1	Phase 1	Phase 1
Chemistry (Sept., 1967)	Phase 1	Phase 1	---	---
Biology (Sept., 1968)	Phase 1	Phase 1	Phase 1	---
Elementary School Mathematics (Sept., 1968)	Phase 1	---	Phase 1	---
	---	Phase 2	---	---
	---	Phase 3	Phase 3	Phase 3
Veterinary Medicine (Sept., 1970)	---	---	Phase 1	Phase 1
Elementary School Reading (April, 1971)	---	Phase 1	---	---
	---	Phase 2	Phase 2	Phase 2
Community College English (July, 1971)	Phase 1	Phase 1	Phase 1	---
	Phase 2	Phase 2	Phase 2	Phase 2
	---	Phase 3	Phase 3	Phase 3

Table 3--Continued

Groups (With Starting Dates)	Models			
	Independent Developer	Colleagueship	Lesson Designer with Programming Assistants	Support Staff
Community College Mathematics (July, 1971)	Phase 1	Phase 1	---	---
	Phase 2	Phase 2	Phase 2	---
Accounting (Sept., 1971)	Phase 1	---	---	---
	---	---	Phase 2	---
Military Training Centers (July, 1972)	---	---	---	Phase 1
Chanute Air Force Base (August, 1972)	---	Phase 1	---	Phase 1
	---	---	Phase 2	Phase 2
Aberdeen Proving Grounds (March, 1973)	---	Phase 1	---	---
Basic Medical Sciences (July, 1973)	---	---	Phase 1	---
	---	Phase 2	Phase 2	Phase 2
*PLATO Services Organization (August, 1973)	---	---	---	Phase 1

Table 3--Continued

Groups (With Starting Dates)	Models			
	Independent Developer	Colleagueship	Lesson Designer with Programming Assistants	Support Staff
**PLATO Educational Evaluation and Research (August, 1973)	---	---	---	Phase 1
Sheppard Air Force Base (Feb., 1974)	---	Phase 1	---	Phase 1

* The people in this group functioned as consultants as early as 1971, but a formal organization was not established until August, 1973.

** Successor to the CERL Evaluation Office, which was begun in August, 1967.

Table 4

Relevant Background Characteristics

	NSF-funded Projects	ARPA-funded Projects	CERL Support Groups	University Groups	Totals
High School Graduates or Students	7	8	0	0	15
College Graduates	13	10	1	5	29
Masters or Equivalent Degrees	27	4	6	7	44
Doctorates	12	4	8	10	34
Teaching Experience	36	16	11	11	74
Curriculum Development Experience	4	4	1	1	10
Non-PLATO Computer Experience	21	5	7	6	39

Table 5

Categories for Collecting Courseware Development Data

	Personnel Time				Equipment Costs		
	Director	Professionals	Junior Staff	Support Staff	Computer Access	Other Equipment	Miscellaneous (Travel, Supplies, etc.)
START-UP:							
Selecting Developers							
Training Others and Consulting							
Being Tranied and Keeping Up							
Exploring System Capabilities							
PLANNING:							
Specifying Content							
Specifying Instructional Strategy							
Specifying Program Structure							
PRODUCTION:							
Instructional Lessons:							
Drill and Practice							
Tutorial							
Games and Simulations							
Other							
Support Lessons:							
Routers							

Table 5--Continued

	Personnel Time				Equipment Costs		
	Director	Professionals	Junior Staff	Support Staff	Computer Access	Other Equipment	Miscellaneous (Travel, Supplies, etc.)
Drivers							
Data Collection							
Other Media: Slides/Microfiche							
Audio Messages							
Workbooks							
Other							
REVIEW: Reviewing Lessons by Other Developers							
Feedback From Other Developers							
Feedback From External Experts							
Feedback From Instructor-Users							
Feedback From Initial Student-Users							
VALIDATION & DOCUMENTATION: Study Design							
Instrument Development							
User Recruitment							

Table 5--Continued

	Personnel Time				Equipment Costs		
	Director	Professionals	Junior Staff	Support Staff	Computer Access	Other Equipment	Miscellaneous (Travel, Supplies, etc.)
Data Collection							
Data Analysis							
Documentation							
IMPLEMENTATION: Maintenance							
Dissemination							
Updating							
HARDWARE/SOFTWARE: Development							
Maintenance							
ADMINISTRATION/COORDINATION							

Appendix

Individual Interview Summary

Name

Curriculum Group

Position
- Appointments
- % time (in each)
- Title on project
- Starting date

Background
- Education
- Experience
  - Previous PLATO work
  - Programming
  - Teaching
  - Curriculum design
- Initial expectations

Product
- What is it (programs, parts of programs, content)?
- Example (if tangible)
- Other products
- Criteria for judging products
- How many instructional hours?
- How many work-hours to produce them?

Role and Process
- Role in production of courseware
- Degree of autonomy
- Nonproduction activities
  - Interaction with instructor-users (demos, workshops, recruiting)
  - Interaction with professional community (conferences, reading/writing publications
- Administration (personnel, coordination of meetings, reports, external evaluation)
- Proctoring of students
- Advising (TUTOR, lesson design)
- Training
- Other (hardware development, etc.)

Inputs Received
- Kinds and sources
- How are they used?
- Review procedures (criteria for acceptance of your work)

Online/offline methods and behavior
- What is done online
- What is done offline

Use of Time
- Time devoted to single tasks (continuous?   interrupted?)
- Partition of work load (uniform?   peaks?)

Ideal Production Model
- Division of roles
- Characteristics of staff
- Other comments

Use of TUTOR
- Learning time and conditions
  - Original
  - Continuing
- Level of skill (check one)
  - Uses all "tools" (common, algorithms, answer processing, data)
  - Uses some "tools"
  - Writes structured (sequenced) lessons
  - Writes linear lessons
  - Typing (some coding)
  - No editing (sign-on and communications only)
- Correspondence between skill level and job requirements
- Kind of programming
  - Instructional
  - Support
  - Revision

- Use of resources/work style
  - Flowcharts
  - Printouts
  - AIDS/manuals
  - Consultation
- TUTOR as aid or hindrance to instructional design

Attitudes
- Motivations and rewards
- Principal dislikes about job
- Long-term impact of PLATO
- Impact of PLATO on individual (benefits, side effects)

Comments